



Prática 6 – Listas Duplamente Encadeadas

Problema 1

Considere as estruturas para implementação de uma lista duplamente encadeada circular, não ordenada.

```
typedef int TipoChave;
typedef struct {
    TipoChave Chave;
    /* outros componentes */
} TipoItem;
typedef struct Celula_str *Ponteiro;
typedef struct Celula_str {
    TipoItem Item;
    Ponteiro esq;
    Ponteiro dir;
} Celula;

typedef struct {
    Ponteiro Primeiro, Ultimo;
    int Tamanho;
} TipoLista;
```

Construa as funções de manipulação desta estrutura de dados (e também uma função *main* para administrar a lista, usando um menu de opções):

- a. Fazer (criar) lista vazia.
- b. Testar se lista está vazia.
- c. Retornar tamanho da lista.
- d. Buscar elemento na lista.
- e. Inserir elemento ao final da lista.
- f. Remover elemento da lista.
- g. Imprimir a lista na tela.

Problema 2

Um autômato móvel é controlado a partir da indicação da direção que ele deve seguir, num plano de duas dimensões, e a distância que deve percorrer. As direções são frente, trás, esquerda e direita. Prepare um programa que permita armazenar, em uma lista dinâmica duplamente encadeada, um percurso completo digitado pelo usuário, composto de vários trechos. Cada trecho é formado por um número único de identificação do trecho, a direção (frente, trás, esquerda, direita) a seguir e a distância a ser percorrida (na unidade apropriada). O número único de identificação do trecho é gerado automaticamente pelo programa, seqüencialmente; os demais dados são digitados pelo usuário, ao construir o percurso.

A seguinte estrutura pode ser usada:

```
typedef struct {
    int idTrecho;
    char direcao;
    float distancia;
} TipoTrecho;
```

Use as funções de manipulação de listas dinâmicas duplamente encadeadas já implementadas, adaptando-as apropriadamente para os novos tipos e estruturas. Na função de impressão do percurso total, adicione o cálculo e impressão da distância total que o autômato foi programado a percorrer, considerando o percurso total digitado.